Εφαρμογή Πρόβλεψης Ύπαρξης Συντρόφου

Σπαθάρας Άγγελος και Μπασδάνης Διονύσιος

Εξόρυξη Δεδομένων 2018-19
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Πανεπιστήμιο Θεσσαλίας, Βόλος
{aspatharas,dbasdanis}@e-ce.uth.gr

1 Οδηγίες Εγκατάτασης

Για να εγκαταστήσουμε την βιβιοθήκη ώστε να χρησιμοποιούμε το Weka μέσω python χρειαζόμαστε το python-weka-wrapper και το javabridge. Αναλυτικότερες οδηγίες θα βρείτε στο link: https://fracpete.github.io/python-weka-wrapper/install.html

2 Οδηγίες Εκτέλεσης

Για να τρέξουμε την εφαρμογή ανοίγουμε μια κονσόλα στον υπολογιστή μας και πηγαίνουμε στο φάκελο που είναι το αρχείο gui.py. Έπειτα εκτελούμε την εντολή »python gui.py.

3 Αρχεία

Άρχεία που χρησιμοποιήθηκαν:
-Το αρχείου answers.csv περιέχει τις απαντήσεις του ερωτηματολογίου.
-Το αρχείο demo.webm είναι ένα βίντεο που δείχνει κάποια παραδείγματα εκτέλεσης της εφαρμογής μας.
-Το αρχείο simplefiedanswers.csv περιέχει τα δεδομένα που τα γνωρίσματα έχουν μικρότερο εύρος τιμών.

4 Ανάπτυξη της εφαρμογής

Για να εξάγουμε το μοντέλο που χρησιμοποιεί η εφαρμογή ακολουθούμε τα εξής βήματα:
1. Ανοίγουμε το Weka.
2. Φορτώνουμε το dataset answers.csv.
3. Αφαιρούμε τα γνωρίσματα Eyes colour και Zodiac sign.
4. Χρησιμοποιούμε τον κατηγοριοποιητή J48.
5. Αποθηκέυουμε το μοντέλο στο αρχείο J48.model.

Στο αρχείο python εισάγουμε τις βιβλιοθήκες για να χρησιμοποιήσουμε το Weka με τον εξής τρόπο:

```
import weka.core.jvm as jvm
import weka.core.serialization as serialization
from weka.core.converters import Loader
from weka.classifiers import Classifier
from weka.core.dataset import Instance
from weka.core.dataset import Instances
from weka.core.dataset import Attribute
```

Και χτίζουμε το μοντέλο μέσω της python με τον εξή τρόπο:

$$'''_{C}reatethemodel'''$$
$$objects = serialization.read_all("J48.model")$$

$$cls = Classifier(jobject = objects[0])$$
$$data = Instances(jobject = objects[1])$$

$$'''Createthetestsettobeclassified'''$$
$$gender_values = ["Man", "Woman"]$$
$$sociability_values = ["Introvert", "Extrovert"]$$
$$stability_values = ["Stable", "Unstable"]$$

$$values = [gender_values.index(gender), age, height, weight, self.BMI(weight, height),$$
$$stability_values.index(stability), sociability_values.index(sociability),$$
$$Instance.missing_value()]$$

$$inst = Instance.create_instance(values)$$
$$inst.dataset = data$$

$$'''Classification''$$
$$prediction = int(cls.classify_instance(inst))$$

5 Αποτελέσματα

Χρησιμποιώντας το αρχείο answers.csv και χωρίς να επεξεργαστούμε τα γνωρίσματα έχουμε τα παρακάτω αποτελέσματα:

ZeroR -> 51.4286 %
NaiveBayes -> 66.4286 %
Logistic -> 63.5714 %
HoeffdingTree -> 67.1429 %
J48 -> 65 %

Έπειτα αφαιρώντας τα γνωρίσματα Eyes colour και Zodiac sign έχουμε τα αποτελέσματα:

NaiveBayes -> 66.4286 %
Logistic -> 67.8571 %
HoeffdingTree -> 67.8571 %
J48 -> 72.1429 %

Τέλος χρησιμοποιώντας το αρχείο simpolefiedanswers.csv παίρνουμε τα εξής:

NaiveBayes -> 66.4286 %
Logistic -> 63.5714 %
HoeffdingTree -> 67.1429 %
J48 -> 73.5714 %

6 Συμπεράσματα

Η μέγιστη απόδοση που μπορεί να φτάσει το σύστημα μας είναι 73.57% και μία μέση απόδοση απο όλους τους αλγορίθμους είνια το 67%. Γενίκα έχουμε ένα ικανοποιητικό αποτέλεσμα σε ένα φαινομενικά τυχαίο set δεδομένων. Ίσως με περισσότερα γνωρίσαμτα η απόδοση μπορεί να γίνει μεγαλύτερη. Περιοριστήκαμε όμως σε αυτά τα δεδομένα γιατί δεν θέλαμε να εμπλακούμε με περισσότερα προσωπικά στοιχεία.